Atitit.java重要知识点attilax总结

[Threadlocal 1](#_Toc11354)

[Classloader 1](#_Toc15703)

[Compiler api 1](#_Toc24211)

[asyn 1](#_Toc28023)

[Concurrent 1](#_Toc23407)

[反射 1](#_Toc10832)

[构建器initialize() 1](#_Toc31042)

[ThinkingInJava index 2](#_Toc2012)

## Threadlocal

## Classloader

## Compiler api

## asyn

## Concurrent

## 反射

## 构建器initialize()

++为我们引入了“构建器”的概念。这是一种特殊的方法，在一个对象创建之后自动调用。Java也沿用了这个概念，但新增了自己的“垃圾收集器”，能在资源不再需要的时候自动释放它们。本章将讨论初始化和清除的问题，以及Java如何提供它们的支持。  
  
4.1 用构建器自动初始化  
对于方法的创建，可将其想象成为自己写的每个类都调用一次initialize()。这个名字提醒我们在使用对象之前，应首先进行这样的调用。但不幸的是，这也意味着用户必须记住调用方法。在Java中，由于提供了名为“构建器”的一种特殊方法，所以类的设计者可担保每个对象都会得到正确的初始化。若某个类有一个构建器，那么在创建对象时，Java会自动调用那个构建器——甚至在用户毫不知觉的情况下。所以说这是可以担保的！  
接着的一个问题是如何命名这个方法。存在两方面的问题。第一个是我们使用的任何名字都可能与打算为某个类成员使用的名字冲突。第二是由于编译器的责任是调用构建器，所以它必须知道要调用是哪个方法。C++采取的方案看来是最简单的，且更有逻辑性，所以也在Java里得到了应用：构建器的名字与类名相同。这样一来，可保证象这样的一个方法会在初始化期间自动调用。

## ThinkingInJava index

IMG_256IMG_257[第1章 对象入门](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter1.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_258IMG_259[第2章 一切都是对象](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter2.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_260IMG_261[第3章 控制程序流程](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter3.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_262IMG_263[第4章 初始化和清除](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter4.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_264IMG_265[第5章 隐藏实施过程](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter5.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_266IMG_267[第6章 类再生](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter6.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_268IMG_269[第7章 多形性](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter7.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_270IMG_271[第8章 对象的容纳](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter8.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_272IMG_273[第9章 违例差错控制](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter9.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_274IMG_275[第10章 Java IO系统](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter10.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_276IMG_277[第11章 运行期类型鉴定](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter11.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_278IMG_279[第12章 传递和返回对象](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter12.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_280IMG_281[第13章 创建窗口和程序片](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter13.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_282IMG_283[第14章 多线程](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter14.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_284IMG_285[第15章 网络编程](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter15.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_286IMG_287[第16章 设计范式](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter16.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

IMG_288IMG_289[第17章 项目](http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/chapter/chapter17.htm" \t "http://www.yq1012.com/ThinkingInJava/Thinking+In+Java/body)

## Effective Java中文版

Line 71: 第1章 引言

Line 72: 第2章 创建和销毁对象

Line 80: 第3章 对于所有对象都通用的方法

Line 86: 第4章 类和接口

Line 97: 第5章 泛型

Line 105: 第6章 枚举和注解

Line 114: 第7章 方法

Line 122: 第8章 通用程序设计

Line 135: 第9章 异常

Line 145: 第10章 并发

Line 154: 第11章 序列化

Line 71: 第1章 引言

Line 72: 第2章 创建和销毁对象

Line 73: 第1条：考虑用静态工厂方法代替构造器

Line 74: 第2条：遇到多个构造器参数时要考虑用构建器

Line 75: 第3条：用私有构造器或者枚举类型强化Singleton属性

Line 76: 第4条：通过私有构造器强化不可实例化的能力

Line 77: 第5条：避免创建不必要的对象

Line 78: 第6条：消除过期的对象引用

Line 79: 第7条：避免使用终结方法

Line 80: 第3章 对于所有对象都通用的方法

Line 81: 第8条：覆盖equals时请遵守通用约定

Line 82: 第9条：覆盖equals时总要覆盖hashCode

Line 83: 第10条：始终要覆盖toString

Line 84: 第11条：谨慎地覆盖clone

Line 85: 第12条：考虑实现Comparable接口

Line 86: 第4章 类和接口

Line 87: 第13条：使类和成员的可访问性最小化

Line 88: 第14条：在公有类中使用访问方法而非公有域

Line 89: 第15条：使可变性最小化

Line 90: 第16条：复合优先于继承

Line 91: 第17条：要么为继承而设计，并提供文档说明，要么就禁止继承

Line 92: 第18条：接口优于抽象类

Line 93: 第19条：接口只用于定义类型

Line 94: 第20条：类层次优于标签类

Line 95: 第21条：用函数对象表示策略

Line 96: 第22条：优先考虑静态成员类

Line 97: 第5章 泛型

Line 105: 第6章 枚举和注解

Line 114: 第7章 方法

Line 115: 第38条：检查参数的有效性

Line 116: 第39条：必要时进行保护性拷贝

Line 117: 第40条：谨慎设计方法签名

Line 118: 第41条：慎用重载

Line 119: 第42条：慎用可变参数

Line 120: 第43条：返回零长度的数组或者集合，而不是：null

Line 121: 第44条：为所有导出的API元素编写文档注释

Line 122: 第8章 通用程序设计

Line 123: 第45条：将局部变量的作用域最小化

Line 124: 第46条：for-each循环优先于传统的for循环

Line 125: 第47条：了解和使用类库

Line 126: 第48条：如果需要精确的答案，请避免使用float和double

Line 127: 第49条：基本类型优先于装箱基本类型

Line 128: 第50条：如果其他类型更适合，则尽量避免使用字符串

Line 129: 第51条：当心字符串连接的性能

Line 130: 第52条：通过接口引用对象

Line 131: 第53条：接口优先于反射机制

Line 132: 第54条：谨慎地使用本地方法

Line 133: 第55条：谨慎地进行优化

Line 134: 第56条：遵守普遍接受的命名惯例

Line 135: 第9章 异常

Line 136: 第57条：只针对异常的情况才使用异常

Line 137: 第58条：对可恢复的情况使用受检异常，对编程错误使用运行时异常

Line 138: 第59条：避免不必要地使用受检的异常

Line 139: 第60条：优先使用标准的异常

Line 140: 第61条：抛出与抽象相对应的异常

Line 141: 第62条：每个方法抛出的异常都要有文档

Line 142: 第63条：在细节消息中包含能捕获失败的信息

Line 143: 第64条：努力使失败保持原子性

Line 144: 第65条：不要忽略异常

Line 145: 第10章 并发

Line 146: 第66条：同步访问共享的可变数据

Line 147: 第67条：避免过度同步

Line 148: 第68条：executor和task优先干线程

Line 149: 第69条：并发工具优先于wait和notify

Line 150: 第70条：线程安全性的文档化

Line 151: 第71条：慎用延迟初始化

Line 152: 第72条：不要依赖于线程调度器

Line 153: 第73条：避免使用线程组

Line 154: 第11章 序列化

Line 155: 第74条：谨慎地实现Serializable接口

Line 156: 第75条：考虑使用自定义的序列化形式

Line 157: 第76条：保护性地编写readObject方法

Line 158: 第77条：对于实例控制，枚举类型优先于readResolve

Line 159: 第78条：考虑用序列化代理代替序列化实例

JAVA核心技术（卷1） (豆瓣).html

第1章 Java程序设计概述

第2章 Java程序设计环境

第3章 Java基本的程序设计结构

第4章 对象与类

第5章 继承

第6章 接口与内部类

第7章 图形程序设计

第8章 事件处理

第9章 Swing用户界面组件

第10章 部署应用程序和applet

第11章 异常、日志、断言和调试

第12章 泛型程序设计

第13章 集合　13.1 集合接口

第14章 多线程

你应该远离的6个Java特性 - 张龙（风中叶）的专栏 - 博客频道 - CSDN.NET.html